Getting the Most out of HPC Networks Using One-Sided Communication

Katherine Yelick

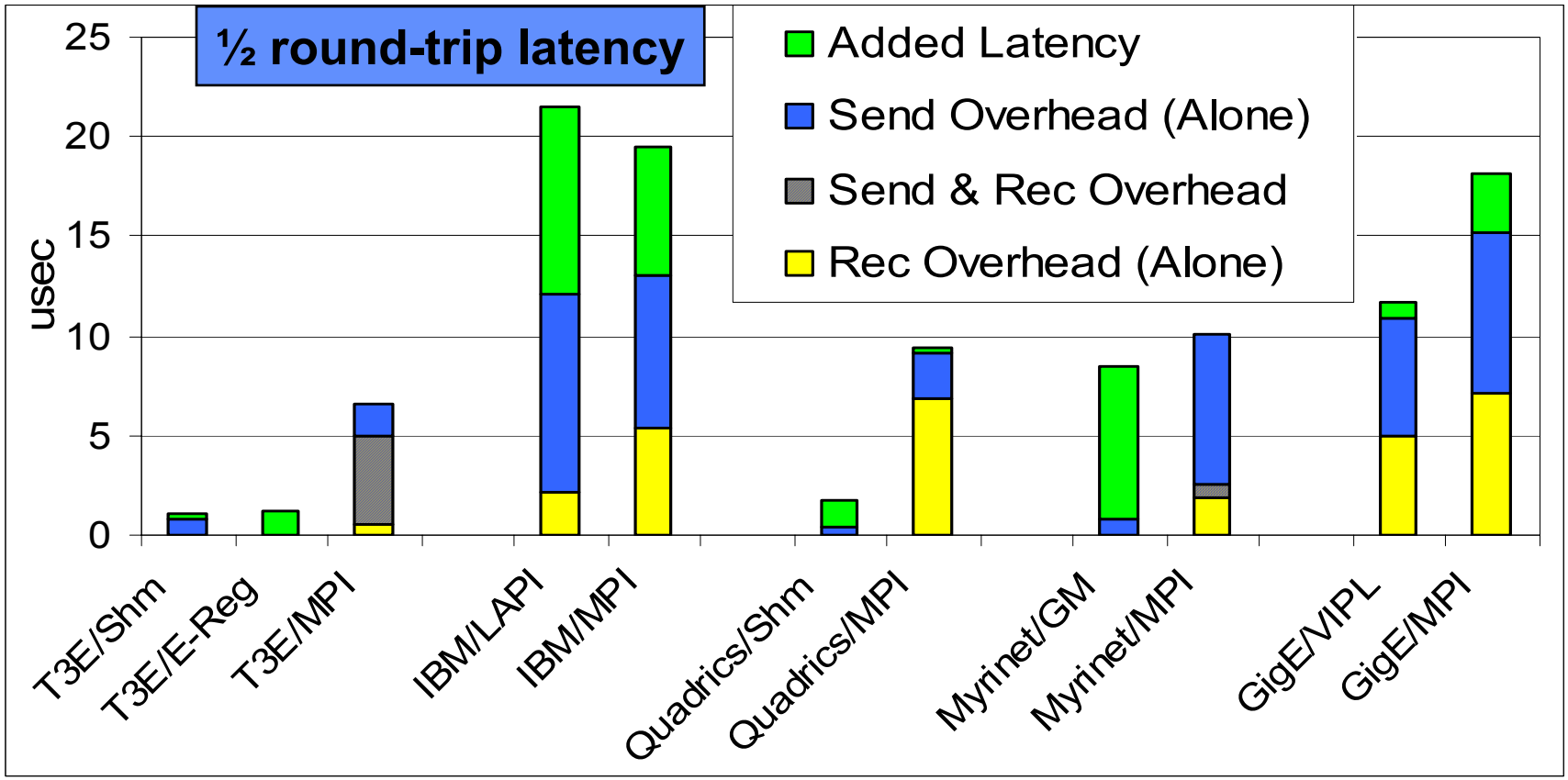
Christian Bell, Rajesh Nishtala, Dan Bonachea

http://upc.lbl.gov

Bisection Bandwidth in HPC Applications

- **Bisection Bandwidth**
  - Bisection bandwidth is the bandwidth across the narrowest part of the network
  - Important in Global transpose operations, exchanges, Alltoall, etc.
- **"Full bisection bandwidth" is expensive**
  - Fraction of machine cost in the network is increasing
  - Fat-tree and full crossbar topologies may be too expensive
  - Especially on machines with 100K and more processors
  - SMP clusters often limit bandwidth at the node level

Historical Perspective



- Potential performance advantage for fine-grained, one-sided programs
- Potential productivity advantage for irregular applications

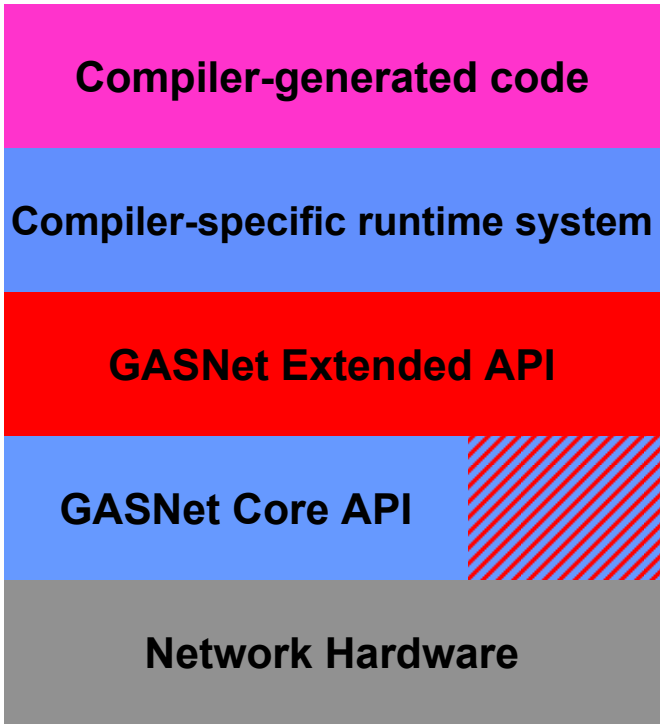
GASNet Communications System

GASNet offers expressive put/get primitives

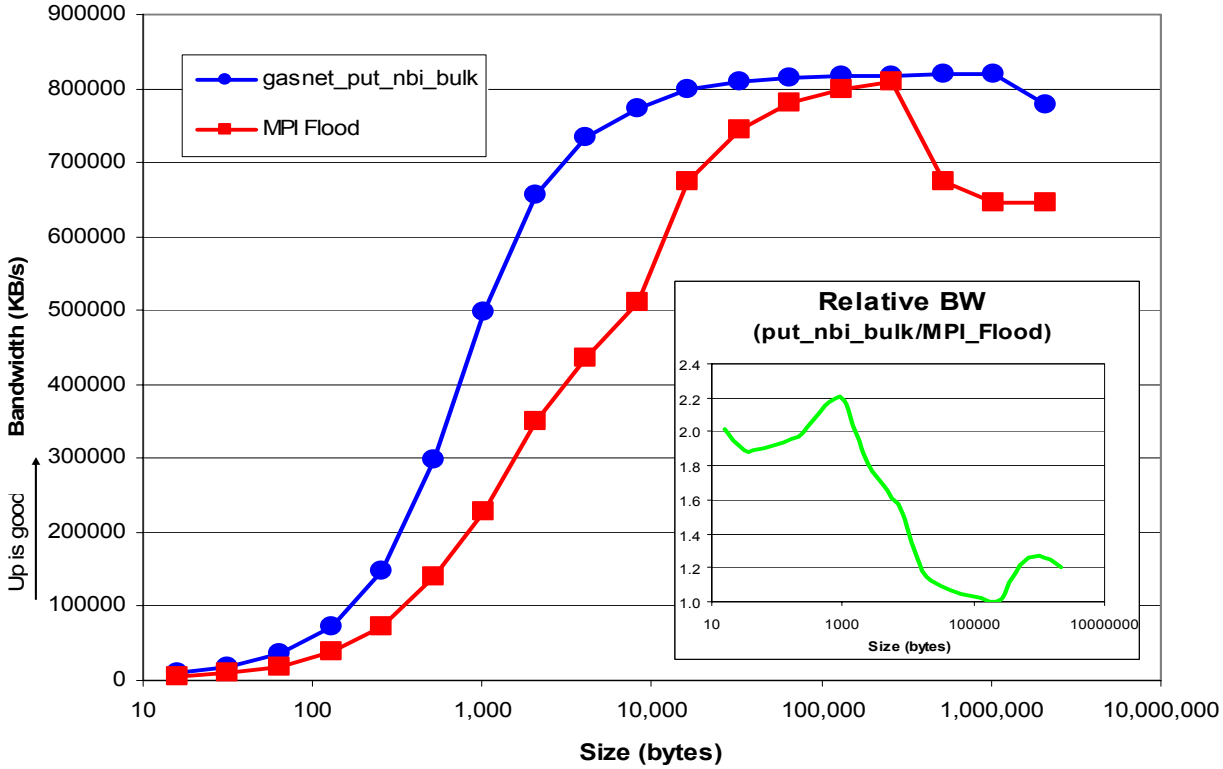
- Contiguous (and recently) non-contiguous communication support
- Communication can be blocking or non-blocking (explicit with handles or implicit globally/region-based)
- Transfers can be memory-to-memory or memory-to-register
- Synchronization can poll or block
- Allows expressing complex split-phase communication (compiler optimizations)

2-Level architecture to ease implementation:

- Core API
  - Based on Active Messages
- Extended API
  - Used to leverage native network support for high-level operations (RDMA put/get)

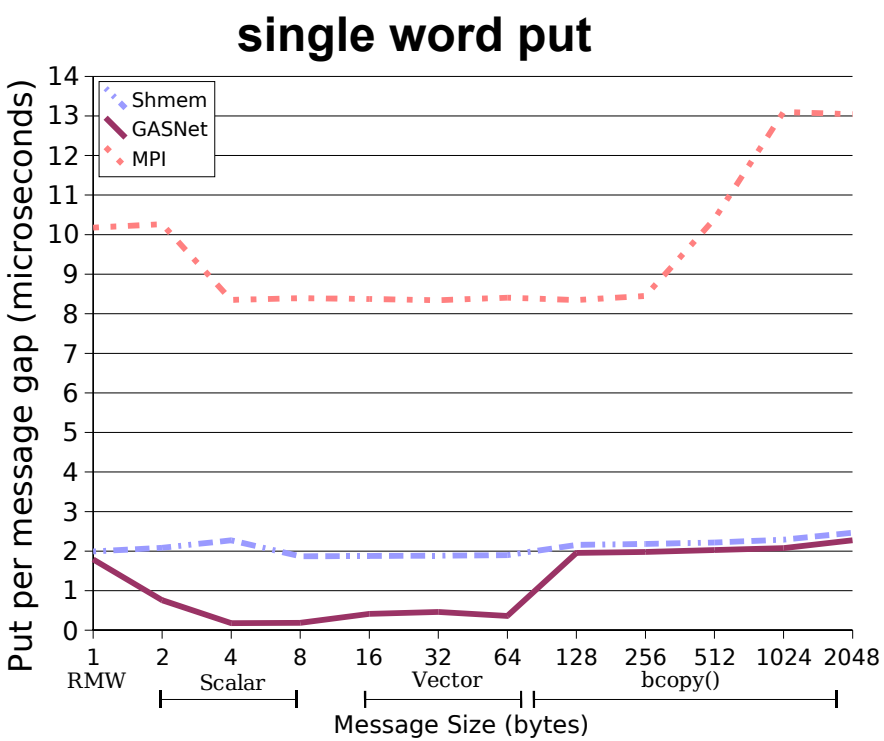


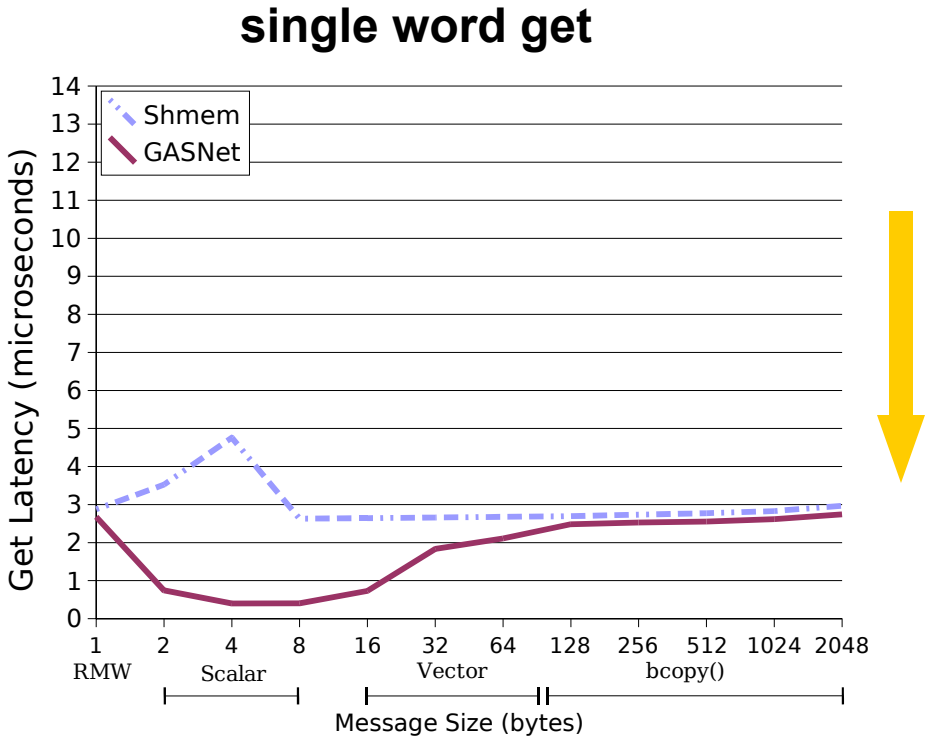
Performance Advantage of One-Sided Communication: GASNet vs 2-Sided MPI



- Comparison on Opteron/InfiniBand – GASNet's vapi-conduit and OSU MPI 0.9.5
- Up to large message size (> 256 Kb), GASNet provides up to 2.2X improvement in streaming bandwidth
- Half power point (N/2) differs by *one order of magnitude*

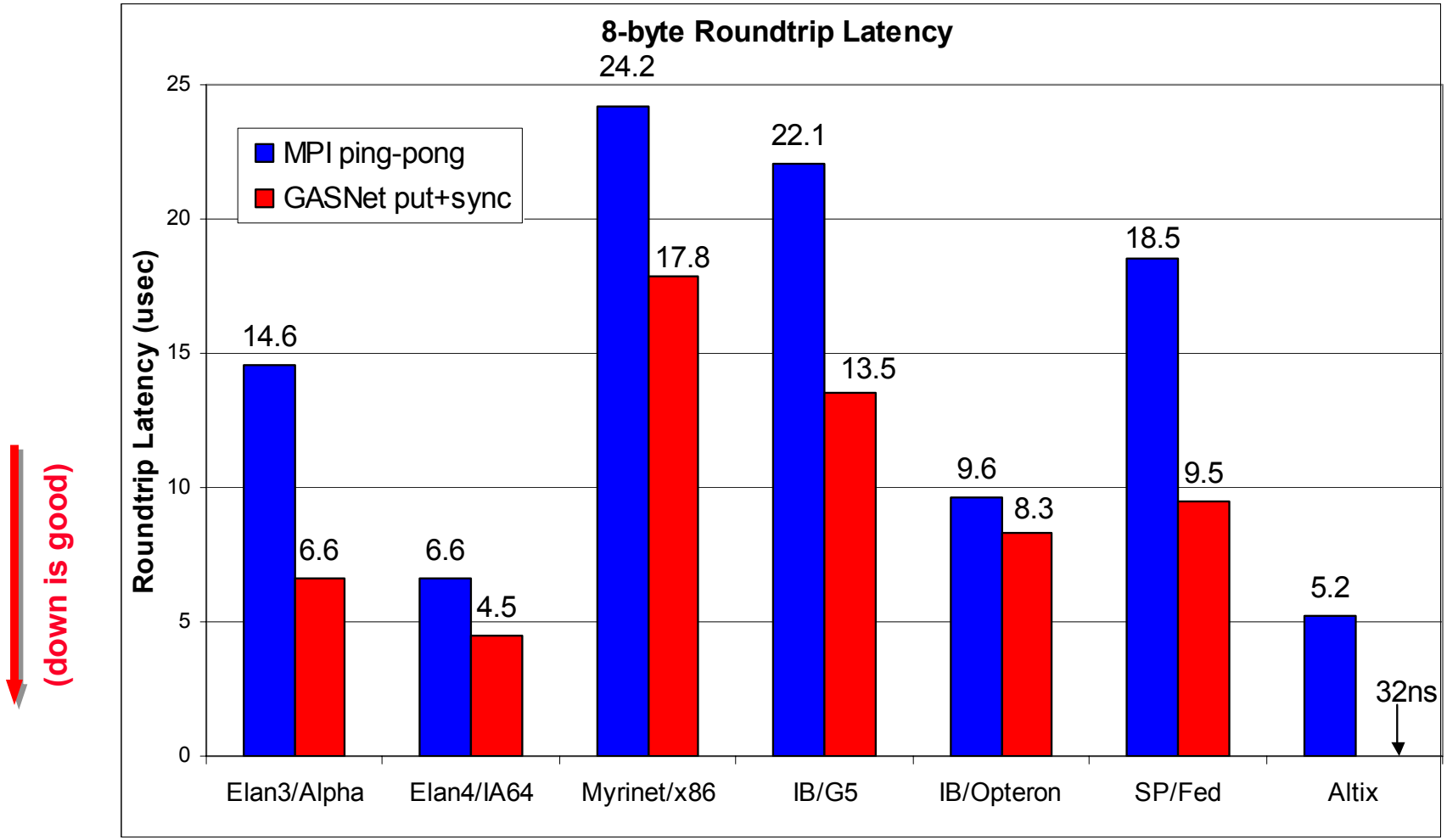
GASNet/X1 Performance





- GASNet/X1 improves small message performance over shmem and MPI
- Leverages global pointers on X1
- Highlights advantage of languages vs. library approach

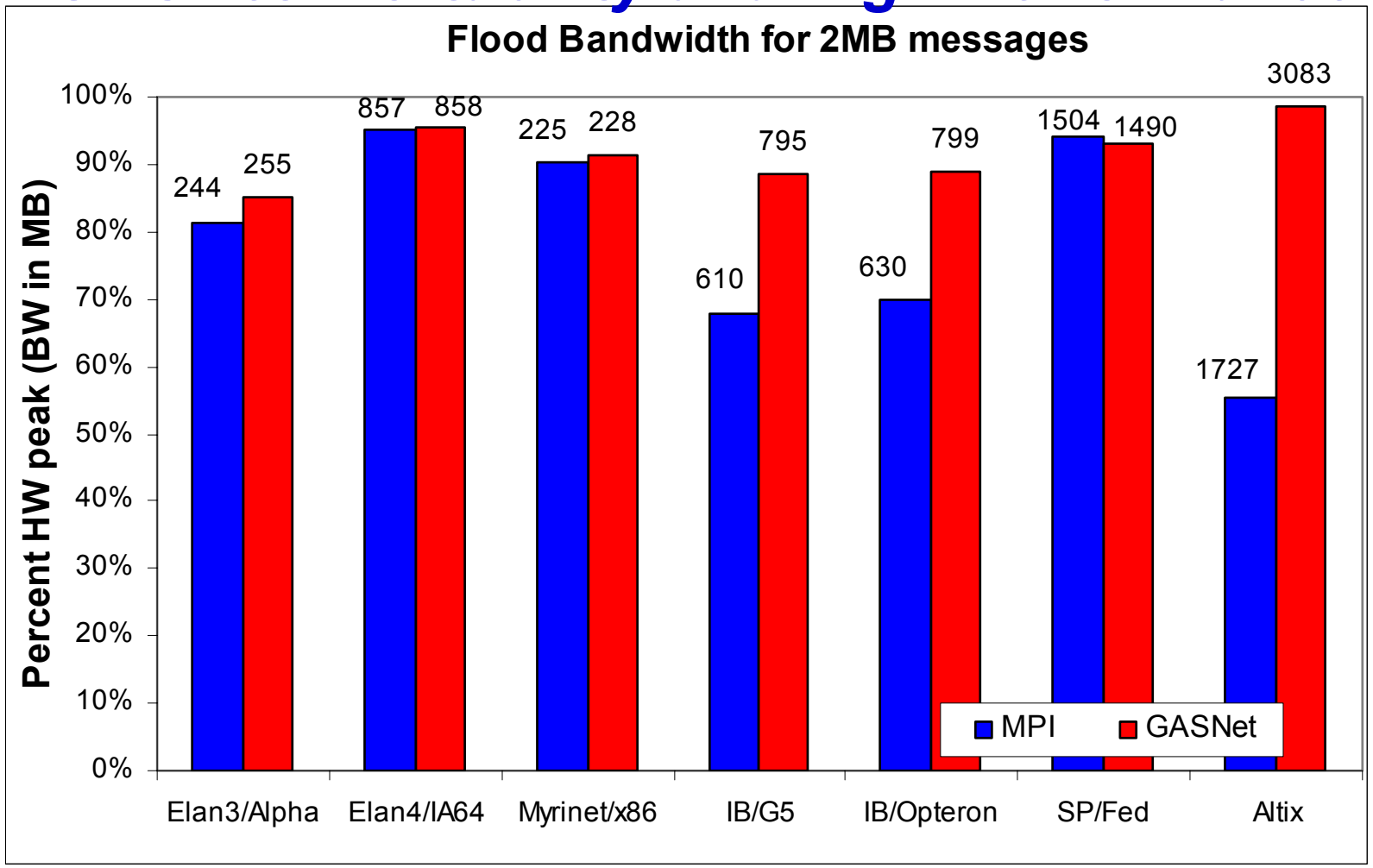
GASNet: Portability and High-Performance



Small-message latency advantage due to RDMA or GAS support

Better RMA support → bigger the win

GASNet: Portability and High-Performance

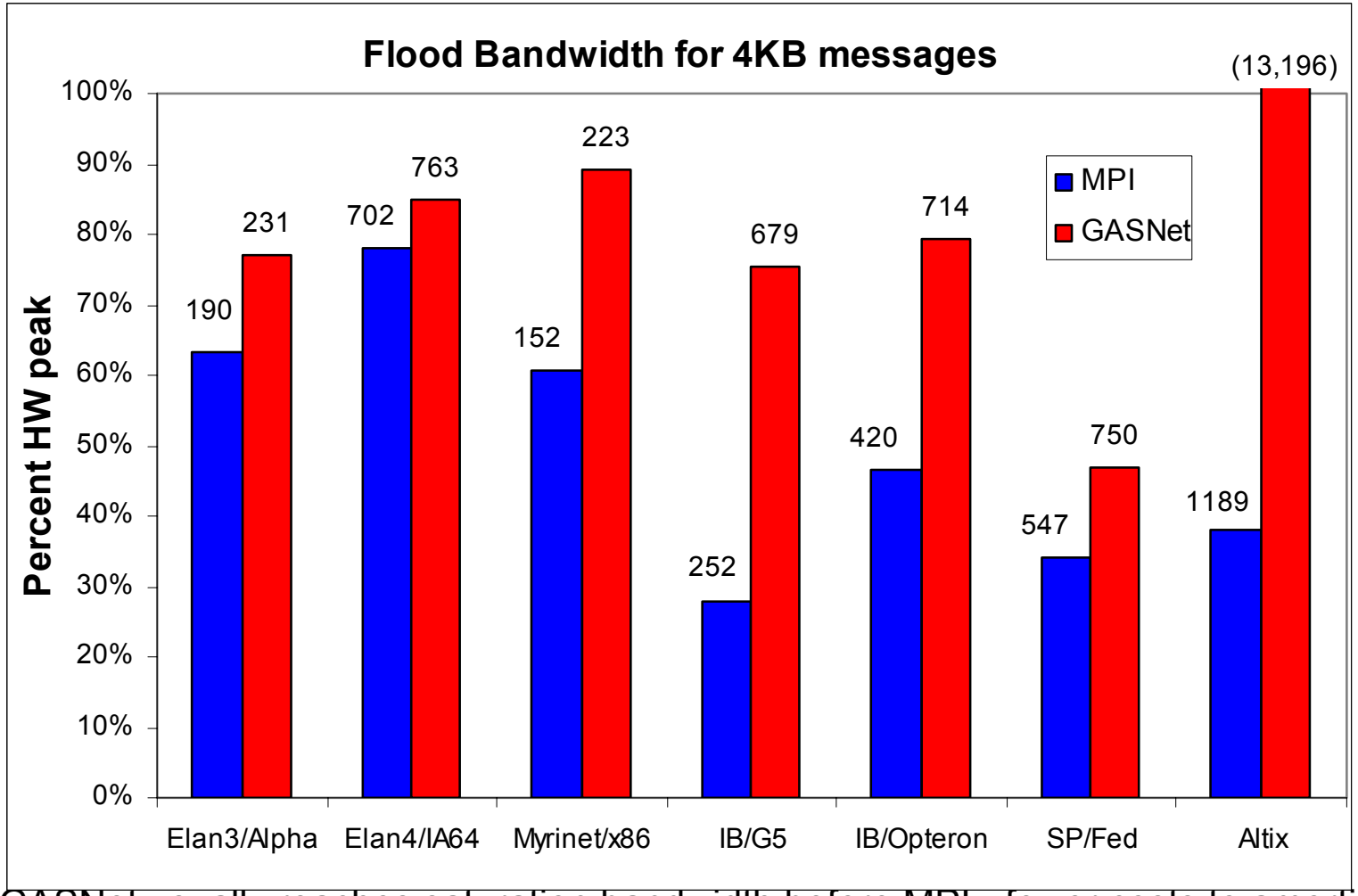


(up is good)

MPI traditionally been tuned for large-message peak bandwidth, GASNet can meet or exceed

In some cases still see a peak B/W advantage to MPI: avoid copies/packetization costs

GASNet: Portability and High-Performance



(up is good)

GASNet usually reaches saturation bandwidth before MPI - fewer costs to amortize

Usually outperform MPI at medium message sizes - often by a large margin

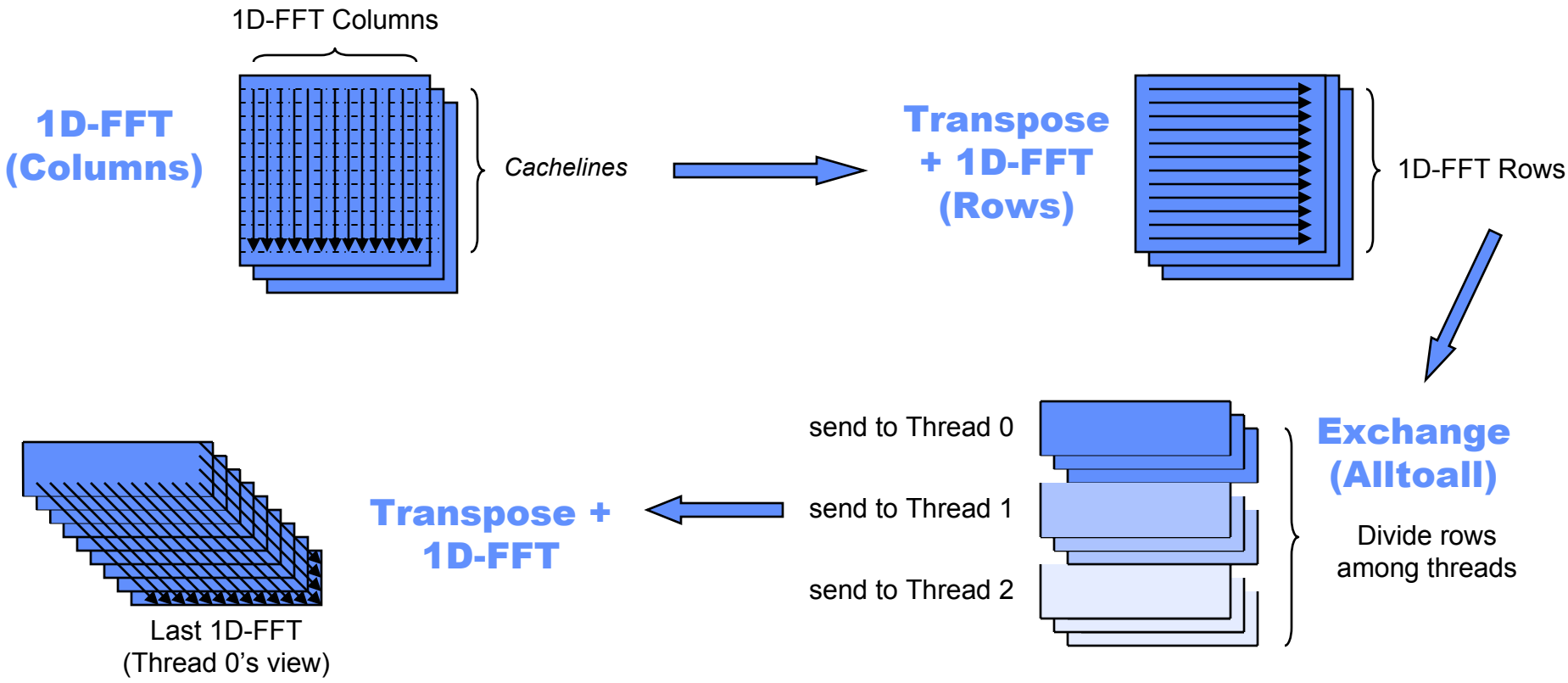
NAS FT Case Study

- **Performance of Exchange (Alltoall) is critical**
  - Communication to computation ratio increases with faster, more optimized 1-D FFTs
  - Determined by available bisection bandwidth
  - Between 30-40% of the applications total runtime
- **Two ways to reduce Exchange cost**
  1. Use a better network (higher Bisection BW)
  2. Overlap the all-to-all with communication (where possible) – "break up" the exchange

  Default NAS FT Fortran/MPI relies on #1

  Our approach uses UPC/GASNet and builds on #2

3D FFT Operation with Global Exchange



- **Single Communication Operation (Global Exchange) sends THREADS large messages**
- **Separate computation and communication phases**

Overlapping Communication

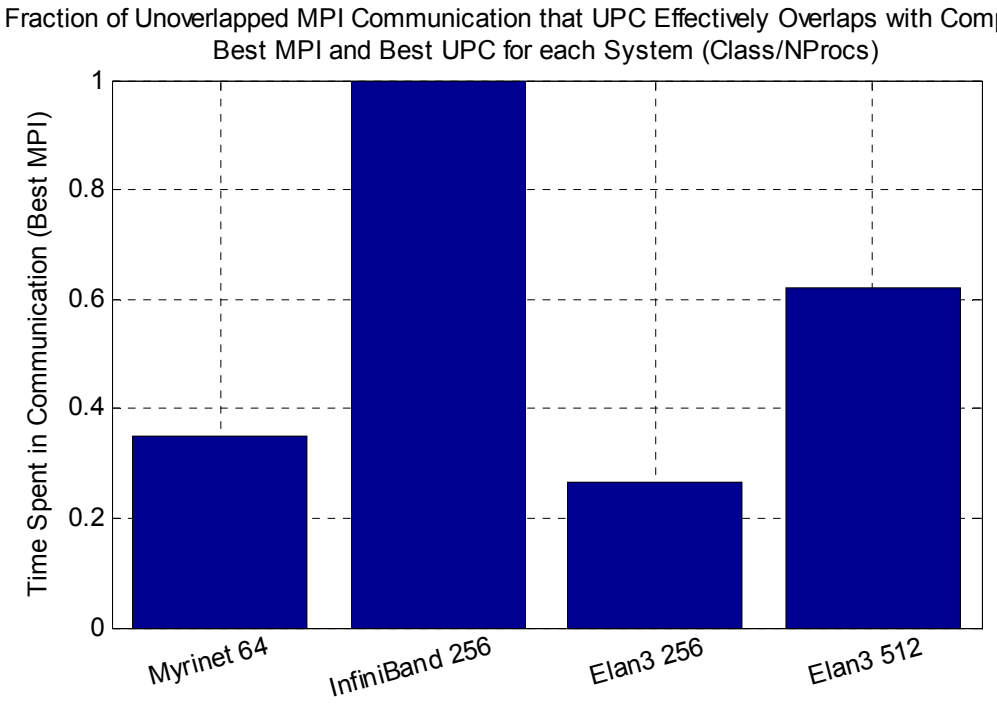
- **Goal: make use of "all the wires"**
  - Distributed memory machines allow for asynchronous communication
  - Berkeley Non-blocking extensions expose GASNet's non-blocking operations
- **Approach: Break all-to-all communication**
  - Interleave row computations and row communications since 1D-FFT is independent across rows
  - Decomposition can be into slabs (contiguous sets of rows) or pencils (individual row)
  - Pencils allow:
    - Earlier start for communication "phase" and improved local cache use
    - But more smaller messages (same total volume)

Decomposing NAS FT Exchange into Smaller Messages

- **Example Message Size Breakdown for Class D at 256 Threads**

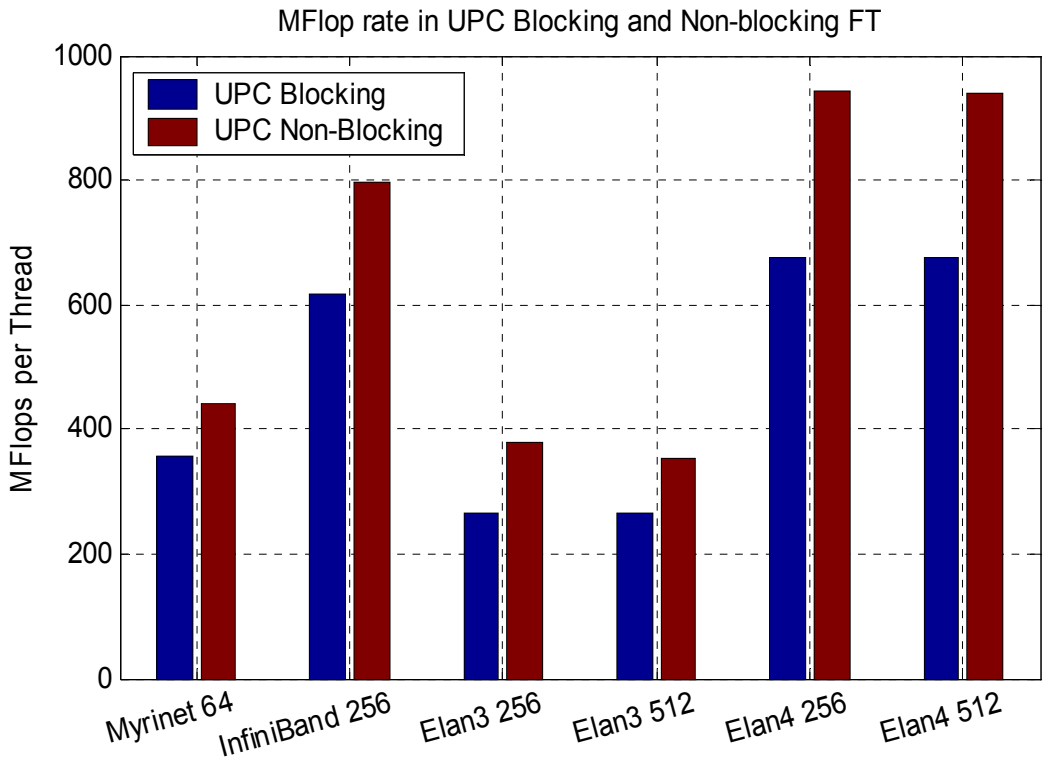
Exchange (Default)	**512 Kbytes**
Slabs (set of contiguous rows)	**65 Kbytes**
Pencils (single row)	**16 Kbytes**

Pencil/Slab optimizations: UPC vs MPI



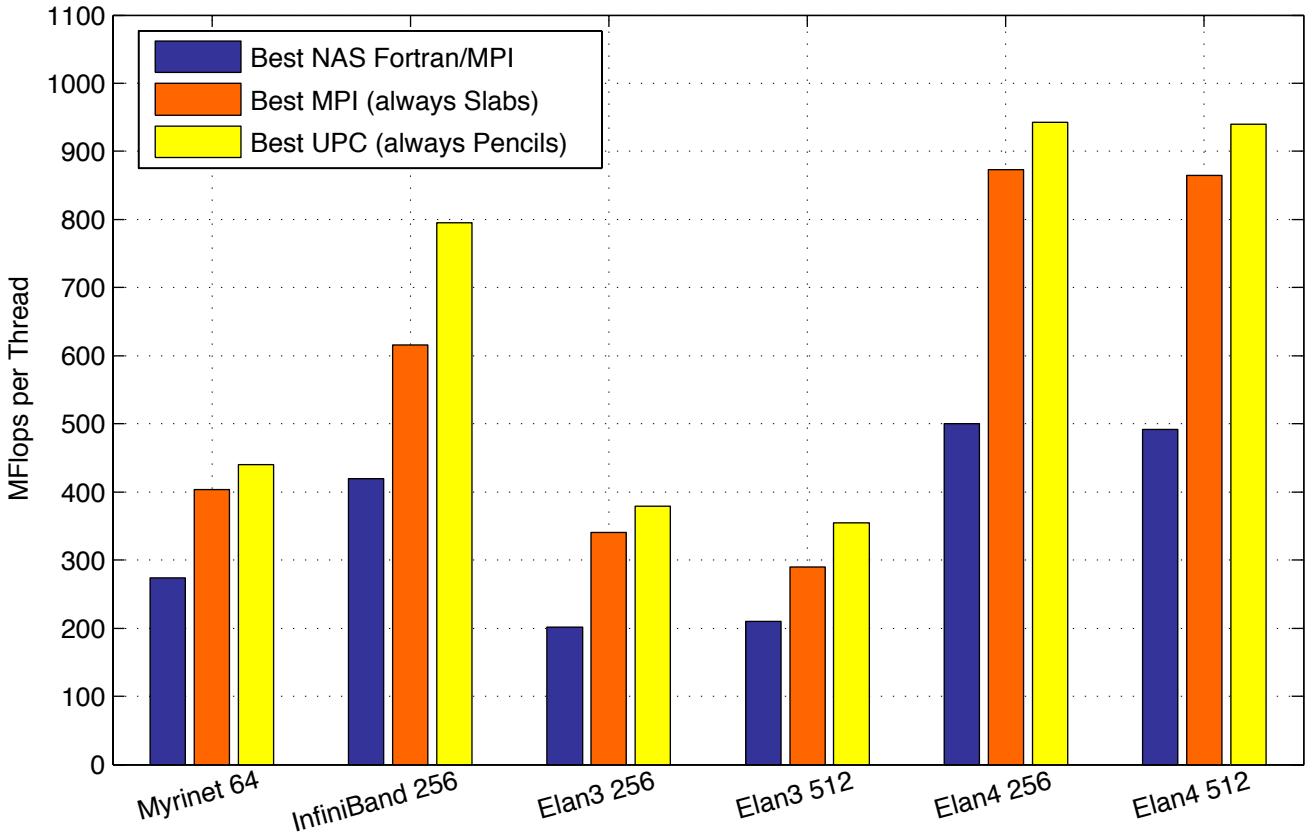
- **Same data, viewed in the context of what MPI is able to overlap**
- **"For the amount of time that MPI spends in communication, how much of that time can UPC effectively overlap with computation"**
- **On Infiniband, UPC overlaps almost all the time the MPI spends in communication**
- **On Elan3, UPC obtains more overlap than MPI as the problem scales up**

NAS FT: UPC Non-blocking MFlops



- **Berkeley UPC compiler support non-blocking UPC extensions**
- **Produce 15-45% speedup over best UPC Blocking version**
- **Non-blocking version requires about 30 extra lines of UPC code**

NAS FT Variants Performance Summary



- **Shown are the largest classes/configurations possible on each test machine**
- **MPI not particularly tuned for many small/medium size messages in flight (long message matching queue depths)**

Summary

- **One-sided communication has performance advantages**
  - Better match for most networking hardware
    - Most cluster networks have RDMA support
    - Machines with global address space support (X1, Altix) shown elsewhere
  - Smaller messages may make better use of network
    - Spread communication over longer period of time
    - Postpone bisection bandwidth pain
  - Smaller messages can also prevent cache thrashing for packing
    - Avoid packing overheads if natural message size is reasonable